

**Microsoft Robotics Developer
Studio - použití senzorů**

**Microsoft Robotics Studio - Use of
Sensors**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2010

.....

Rád bych poděkoval vedoucí mé bakalářské práce paní RNDr. Elišce Ochodkové, Ph.D za vedení a všem kteří mě při práci podporovali.

Abstrakt

Bakalářská práce se zabývá senzory pro stavebnici Lego Mindstorms NXT. Důraz je kladen na prozkoumání vlastností senzorů a jejich použití v programovacích jazycích NXT-G, RobotC a Microsoft Visual Programming Language. V rámci práce byly naimplementovány ukázkové příklady použití senzorů.

Klíčová slova: Lego NXT, NXT-G, RobotC, VPL, HiTechnic

Abstract

This Bachelor thesis deals with sensors for Lego Mindstorms NXT kit. Emphasis is on sensor features and their use in programming languages NXT-G, RobotC and Microsoft Visual Programming Language. In this thesis there are example implementations of sensor usage.

Keywords: Lego NXT, NXT-G, RobotC, VPL, HiTechnic

Seznam použitých zkratk a symbolů

IR	– Infra Red
AC	– Alternating Current
DC	– Direct Current
LED	– Light-Emitting Diode
USB	– Universal Serial Bus
RAM	– Random Access Memory
RDS	– Microsoft Robotic Developer Studio
VPL	– Microsoft Visual Programing Language
RFID	– Radio Frequency Identification
I2C	– Inter-Integrated Circuit

Obsah

1	Úvod	3
2	Lego NXT	4
2.1	NXT řídicí kostka	4
2.2	Motory	5
3	Vývojová prostředí	6
3.1	NXT-G	6
3.2	RobotC	8
3.3	Microsoft Visual Programming Language	9
4	Senzory	12
4.1	Lego	12
4.2	HiTechnic	16
4.3	Příslušenství	23
4.4	Porovnání senzorů	24
5	Příklady použití	27
5.1	Světelný senzor	27
5.2	IR vyhledávač, IR míč	29
5.3	EOPD, Zvukový a Ultrazvukový senzor	30
5.4	Akcelerometr a Dotykový senzor	30
5.5	Senzor barev a Kompas	31
6	Závěr	32
7	Reference	33
	Přílohy	33
A	Obsah CD	34
B	Zdrojové kódy	35
B.1	Akcelerometr a dotykový senzor	35

Seznam obrázků

1	NXT řídící kostka	5
2	NXT motor	5
3	Prostředí NXT-G	6
4	Bloky NXT-G	7
5	RobotC debugger	8
6	Příklad programu ve VPL	10
7	Bloky VPL	11
8	Dotykový senzor	12
9	Ultrazvukový senzor	13
10	Zvukový senzor	13
11	Světelný senzor	14
12	Lego Senzor barev	14
13	Tepelný senzor	15
14	Akcelerometr	16
15	Osy měření akcelerometru	16
16	Gyroskopický senzor	17
17	Detekční rovina gyroskopického senzoru	17
18	HiTechnic senzor barev V2	18
19	Hodnoty barev	18
20	Kompas	20
21	EOPD	21
22	IR vyhledávač V2	22
23	Hodnoty směru IR vyhledávače	23
24	Infračervený míč	24
25	Model na test oblasti detekce	25
26	Model na test rychlosti čtení	26
27	Sledování čáry NXT-G	27
28	Sledování čáry VPN	28
29	Příklad na použití kompasu a detektoru barev	31

1 Úvod

Technický pokrok dnešních dnů pronikl a stále proniká do všech oblastí všedního života. Mezi jinými odvětvími zažívá velký rozmach i robotika. Spolu s tímto rozvojem se ukazuje, že činností, ve kterých byl člověk donedávna považován nenahraditelným, ubývá. Spolu s tímto vývojem však vznikla nová potřeba vzdělávání. V reakci na tento trend přišla společnost Lego na trh se stavebnicí Lego Mindstorms NXT, která v duchu myšlenky nejznámějšího pedagoga všech dob Jana Amose Komenského "škola hrou", otvírá dvířka do říše robotiky a programování již dětem od 10 let. Tato hračka však není určena pouze do rukou dětí. Může být vynikajícím způsobem použita i jako vzdělávací pomůcka pro dospělé.

Cílem této práce je prozkoumat a popsat možnosti práce se senzory pro Lego Mindstorms NXT a jejich vlastnosti. Dále také u těchto senzorů prozkoumat podporu v programovacích jazycích NXT-G, RobotC a VPL.

Ve 2. kapitole je popsáno, co to je Lego Mindstorms NXT a z čeho se skládá. Kapitola 3. rozebírá jednotlivé programovací jazyky z uživatelského hlediska. 4. kapitola obsahuje popis samotných senzorů Lego a HiTechnic, jejich vlastnosti, využití a podpory v jednotlivých jazycích. Práci zakončuje 5. kapitola příklady použití.

2 Lego NXT

Lego Mindstorms NXT je stavebnice, která umožňuje skládat a programovat roboty. Je určena pro děti od 10 let, ale velký úspěch slaví i u dospělých. Díky tomuto úspěchu existuje mnoho internetových stránek a blogů věnovaných Legu NXT. Není proto nouze o originální modely. Lze například poskládat roboty jako je robot řešící Rubikovu kostku, robot s funkcí Turingova stroje, robot-plotter a další.

V současné době jsou na trhu dvě varianty této stavebnice Lego Mindstorms NXT 1.0 a Lego Mindstorms NXT 2.0. Existuje ale také varianta Education, která je určena pro školy.

Sada Lego Mindstorms NXT 1.0 obsahuje:

- 619 dílků Lego Technic
- NXT řídicí kostku
- Tři krokové servomotory
- Ultrazvukový senzor
- Zvukový senzor
- Světelný senzor
- Dotykový senzor

Stavebnice NXT 2.0 je ochuzena o světelný senzor a zvukový senzor, ale má navíc senzor barev a dotykový senzor. Liší se také ve složení dílků.

2.1 NXT řídicí kostka

Řídicí kostka je ovládací centrum NXT robota a musí tedy být zakomponována do každé konstrukce. Do portů označených písmeny A,B,C se kabely připojují motory a do portů označených čísly 1-4 senzory, viz. obrázek 1. Pomocí softwaru kostky, který se ovládá čtyřmi tlačítky, se dají otestovat Lego senzory a naprogramovat jednoduché aplikace.

Připojení tohoto "mozku" robota k počítači se provádí pomocí USB nebo bluetooth. Přes bluetooth se k robotovi dá připojit také pomocí mobilního telefonu nebo jiné NXT kostky.

Technické parametry:

- 32-bitová ARM7 mikro řídicí jednotka
- 256Kbytes FLASH, 64Kbytes RAM
- 8-bitová AVR mikro řídicí jednotka



Obrázek 1: NXT řídicí kostka

- 4Kbytes FLASH, 512Byte RAM
- bluetooth bezdrátová komunikace (bluetooth třída II kompatibilní V2.0), port USB 2.0
- 4 vstupní porty, 6-drátová kabelová digitální platforma (jeden port obsahuje IEC 61158 typ 4/EN 50 170 kompatibilní rozšířený port pro budoucí použití)
- 3 výstupní porty, 6-drátová kabelová digitální platforma
- LCD grafický displej 100 x 64 pixelů
- Reprodukční - zvuková kvalita 8kHz. Zvukový kanál s 8-bitovým rozlišením a frekvencí 2-16KHz
- Napájení: 6 AA baterií nebo lze dokoupit akumulátorovou baterii

2.2 Motory

Stavebnice je vybavena servomotory (obr. 2). Každý servomotor má vestavěný rotační senzor, který zajišťuje přesné pohyby robotů. Rotační senzor měří otáčení motoru ve stupních nebo celkové otáčení (s přesností $\pm 1^\circ$). Jedno otočení odpovídá 360 stupňům, takže pokud se motor nastaví na otočení o 180 stupňů, provede jeho hřídel půl otáčky [6].



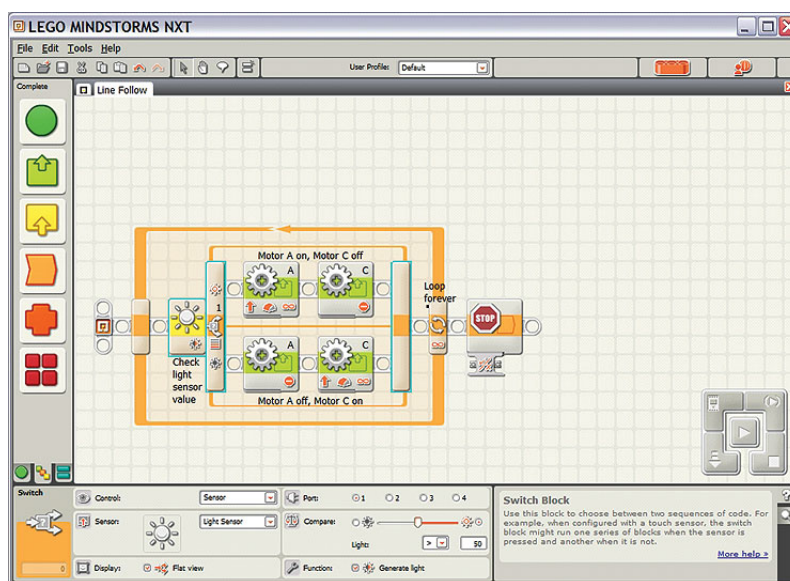
Obrázek 2: NXT motor

3 Vývojová prostředí

Díky tomu, že Lego uvolnilo firmware k NXT kostce, umožnilo tvorbu různých vývojových prostředí. Mezi, které například patří: NQC, NBC a NXC (Jazyky C pro RCX a NXT), Assembler pro NXT, LeJOS (Java pro RCX a NXT), Microsoft Robotics Studio a další[4].

3.1 NXT-G

Prostředí NXT-G se dodává jako součást stavebnice. NXT-G programovací jazyk dostal jméno z programovacího jazyka využívaného programem LabVIEW, vyvinutého firmou National Instruments, který se jmenuje pouze G. Zkratka "G" pochází z faktu, že programovací jazyk je grafický. NXT-G je výsledkem práce firem Lego a National Instruments a je základním programovacím nástrojem pro Lego Mindstorms NXT. Důraz je u NXT-G kladen především na intuitivnost a jednoduchost vývojového prostředí včetně procesu programování tak, aby s robotem mohli pracovat už i žáci základních škol, kteří nemají s programováním žádné nebo minimální zkušenosti. Na obrázku 3 je vidět vzhled prostředí. Největší oblast zabírá plocha, na které programujeme. Na levé straně okna se nachází seznam bloků. Dole jsou detaily právě označeného bloku.

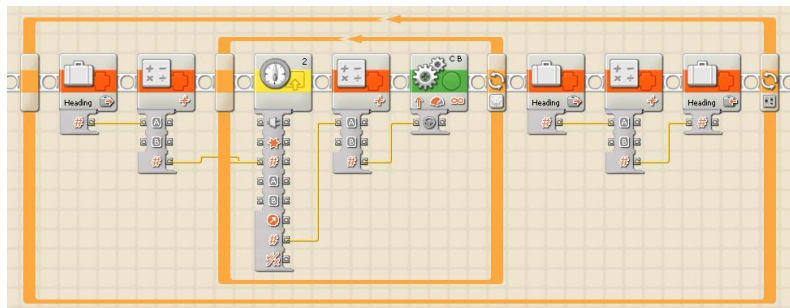


Obrázek 3: Prostředí NXT-G

Programování

Programy v NXT-G se nepíší textem, ale jsou poskládány z ikon, které reprezentují jednotlivé programové instrukce. Těmito instrukcím se říká programovací bloky. Požadované bloky se přetáhnou z panelu nástrojů na pracovní plochu, kde se pospojují v pořadí, v

jakém chceme, aby se prováděly instrukce. Bloky si mezi sebou mohou předávat hodnoty pomocí datových drátů. Kliknutím na blok můžeme zjistit jeho bližší konfiguraci. Vzhled bloků a datových drátů můžeme vidět na obrázku 4. Červené bloky jsou operace s daty, žluté senzory a zelené motory. Datové dráty přenášející data mezi bloky, spojují porty jednotlivých bloků.



Obrázek 4: Bloky NXT-G

Podpora senzorů

Toto vývojové prostředí má plnou podporu Lego senzorů, protože je dodáváno zároveň se stavebnicí. Další bloky senzorů lze přidávat pouze při použití prostředí NXT-G v2.0 nebo u NXT-G v1.0 s updatem. U posledně zmiňovaného, ale výrobce stáhl update ze svých stránek, a proto je nutné mít co nejnovější verzi prostředí. Bloky senzorů HiTechnic najdeme na stránkách jejich výrobce[2].

Výhody

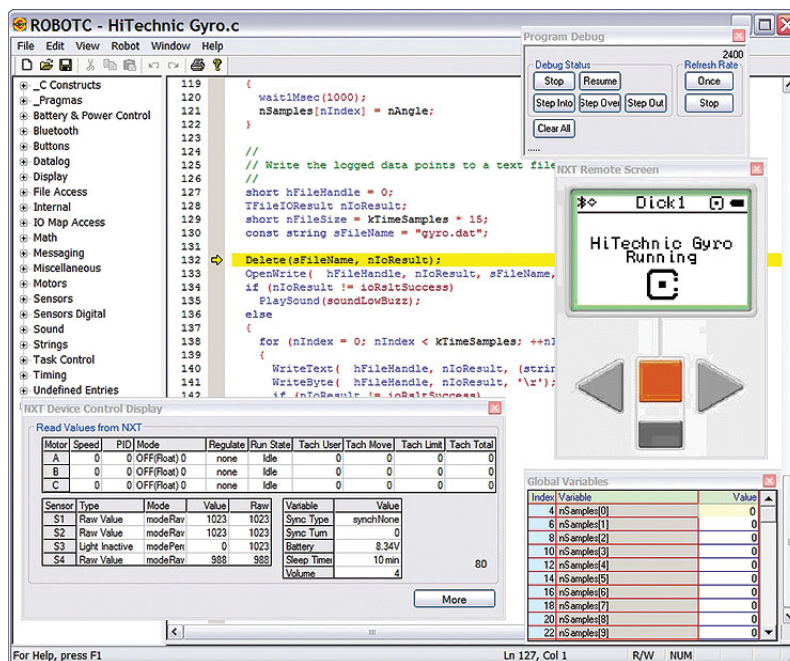
- Jednoduchost
- Dálkové ovládání robota pomocí bluetooth

Nevýhody

- Velikost programů - NXT-G programy mohou být několikanásobně větší oproti jiným prostředím
- Prostředí je hardwarově náročné, nemusí jet plynule a při práci se složitými programy může být nestabilní
- Absence desetinných čísel
- Chybí debugger

3.2 RobotC

RobotC je vývojové prostředí pro programování robotů vytvořené na univerzitě Carnegie Mellon, které je zaměřené na vzdělávání na základních a středních školách. Carnegie Mellon je světová špička v robotickém výzkumu, a to se odráží i v tomto prostředí.



Obrázek 5: RobotC debugger

RobotC není určeno jenom pro NXT, ale také pro další robotické sady jako je VEX a TETRIX. RobotC používá svůj vlastní NXT firmware, který je několikanásobně rychlejší než klasický Lego firmware. Oproti NXT-G umí pracovat s desetinnými čísly a náročnějšími matematickými operacemi. Bohužel má tu nevýhodu, že je licencované.

Programování

Výše zmiňované prostředí RobotC se syntaxemi podobá jazyku C, takže je narozdíl od NXT-G vhodné pro pokročilejší výuku programování. RobotC má velice propracovaný debugger (obr. 5), který umožňuje sledovat všechny proměnné v průběhu programu a funguje i přes bluetooth. Většinu okna zabírá textový editor, kde se píše kód. Na levé straně je seznam funkcí a atributů, které jdou přetáhnout do kódu.[5].

```
const tSensors lightSensor = (tSensors)S1;

task main()
{
    //robot jede dopředu dokud nepřejede černou caru
    while(SensorValue(lightSensor) > 43)
    {

        motor[motorC] = 50;
        motor[motorB] = 50;

    }

    motor[motorC] = 0;
    motor[motorB] = 0;
    wait1Msec(2000);
}
```

Výpis 1: Příklad RobotC programu

Podpora senzorů

Ovládače senzorů Lego a HiTechnic jsou v nejnovějších verzích součástí instalace. Ke starším verzím jsou ke stažení na stránkách HiTechnic.[2] Podpora Senzoru barev Lego není v RobotC v současné době k dispozici.

Výhody

- Přehlednost
- Propracovaný debugger
- Rychlost programů
- Podpora senzorů
- Dokumentace

Nevýhody

- Není zdarma

3.3 Microsoft Visual Programming Language

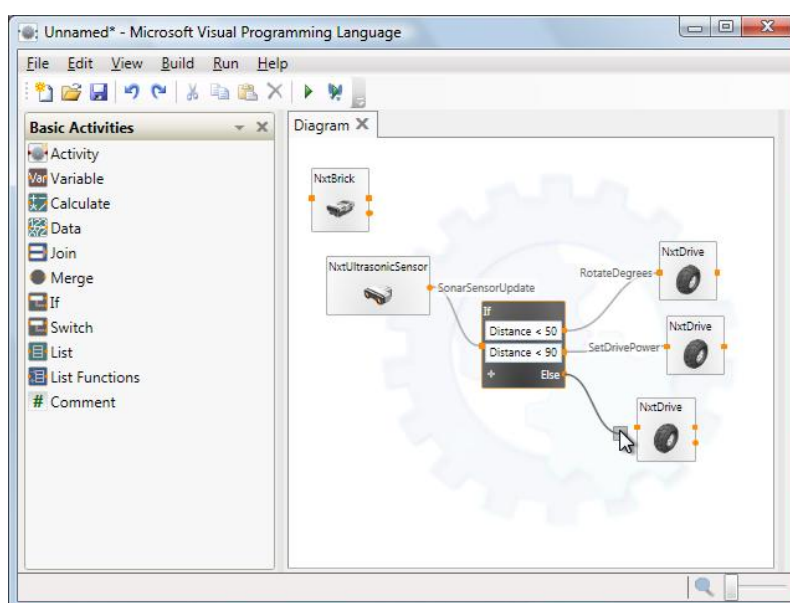
Microsoft Visual Programming Language je prostředí pro vývoj aplikací založený na grafickém zobrazení toku dat (data-flow) a je součástí Microsoft Robotic Developer Studio.

VPL je určeno pro programátory začátečníky se základními znalostmi jako jsou proměnné a logika. Nicméně VPL není jenom pro nováčky. Zkušenější programátoři

jej mohou využít pro rychlý vývoj aplikací. Přestože je VPL primárně určeno pro vývoj programů pro roboty, lze jej využít i k tvorbě jiných aplikací, čímž je VPL vhodné pro širokou škálu uživatelů. VPL programy jdou převést do C#.

VPL nepodporuje většinu NXT senzorů třetích stran jako je i HiTechnic. K tomu abychom tyto senzory mohli používat, musíme mít ve VPL službu, která je ovládá. Služby pro senzory HiTechnic se vyvíjí jako opensource na serveru Codeplex, kde se také dají získat. Je i návod na tvorbu vlastních služeb[8].

VPL program se nenahrává na NXT kostku, jak tomu je u RobotC a NXT-G, ale běží na počítači a přes bluetooth ovládá robota. To s sebou, ale přináší problémy. Komunikace mezi NXT a počítačem má určité zpoždění a data přes bluetooth se mohou ztrácet. Proto není toto prostředí vhodné pro aplikace, které se potřebují rychle rozhodovat a které jsou citlivé na ztrátu dat.



Obrázek 6: Příklad programu ve VPL

VPL data-flow se skládá z posloupnosti spojených aktivit reprezentovaných bloky se vstupy a výstupy (obr 6). Bloky se dají spojovat s jinými bloky. Aktivity mohou představovat již vytvořené služby, data-flow ovládače, funkce nebo jiné moduly. Výsledná aplikace je proto sekvence oddělených procesů.

Aktivity se také mohou skládat z více jiných aktivit. To umožňuje spojit aktivity a použít toto spojení jako blok. V tomto smyslu je aplikace ve VPL také aktivitou. Bloky aktivit obvykle obsahují jméno aktivity a spojovací body. Blok může také obsahovat obrázek, který ukazuje účel aktivity.

Aktivity jsou spojeny přes jejich spojovací body. Bod na levé straně je pro vstupní zprávu a bod na pravé straně pro výstupní zprávu (obr. 7) [9].



Obrázek 7: Bloky VPL

Výhody

- Přehlednost
- Jednoduchost

Nevýhody

- Zpoždění
- Malá podpora senzorů

4 Senzory

Senzory umožňují robotu vnímat prostředí, ve kterém se nachází. Existuje široká škála senzorů pro NXT od kalorimetru, detektoru obsahu kyslíku v kapalinách, RFID, PH senzory po sady pro domácí vytvoření senzorů. Většinu senzorů nevyrábí firma Lego.

4.1 Lego

Tyto senzory jsou součástí sady Lego Mindstorm NXT a proto jsou podporovány ve všech vývojových prostředích. Jejich funkčnost však může být závislá na verzi prostředí. Jedinou výjimkou je Teploměr, který je bez aktualizace podporován pouze v Educational verzi NXT-G v2.0.

4.1.1 Dotykový senzor



Obrázek 8: Dotykový senzor

Dotykový senzor (obr. 8) funguje jako spínač. Vrací tedy true nebo false v závislosti na tom jestli je stisknutý nebo uvolněný.

Použití Signalizace dotyku, přiblížení, detekce překážek, spínač pro uživatelem naprogramovanou akci.

4.1.2 Ultrazvukový senzor

Ultrazvukový senzor (obr. 9) umožňuje detekci objektů, pohybu a měření vzdálenosti. Vzdálenost měří v centimetrech a to od 0 do 255 centimetrů s přesností ± 4 cm. Přesnost klesá se vzdáleností. U vzdáleností nad 50 cm bývá odchylka okolo 1 cm, ale pokud vzdálenost klesne pod 20 cm může odchylka být i 4 cm.

Senzor funguje na podobném principu jako netopýří echolokace. Vyšle ultrazvukový signál a čeká, než se po odrazu vrátí. Podle doby, po kterou signál cestoval, určí vzdálenost. Nejlépe funguje senzor s velkými objekty s pevným povrchem. Může mít problém detekovat překážky, které jsou z látky, zaoblené, tenké nebo malé.



Obrázek 9: Ultrazvukový senzor

V jedné místnosti bychom neměli používat dva nebo více ultrazvukových senzorů. Jejich signály by se rušily a senzory by vracely špatné hodnoty. Tomu lze předejít tím, že roboty oddělíme dostatečně vysokou stěnou.

Použití Vyhýbání se překážkám, měření vzdálenosti, detekce pohybu.

4.1.3 Zvukový senzor



Obrázek 10: Zvukový senzor

Zvukový senzor (obr. 10) dokáže měřit akustický tlak, a to jak v decibelech (dB), tak i v přizpůsobených decibelech (dBA):

- **dBA:** Při detekci přizpůsobených decibelů je citlivost senzoru nastavena tak, ať odpovídá citlivosti lidského ucha. Jinými slovy reaguje pouze na zvuky, které jsme schopni slyšet.
- **dB:** U detekce normálních (nepřizpůsobených) decibelů jsou všechny zvuky měřeny stejně a to i ty, které jsou moc vysoké nebo nízké, na to aby je člověk zaslechl.

Zvukový senzor dokáže měřit akustický tlak až do 90dB, což je přibližně hlasitost závodního motocyklu při maximálním zrychlení. Úroveň akustického tlaku udává senzor v procentech. Například 4-5% je tichá místnost, 5-10% lidé mluvící dál od senzoru, 10-30% rozhovor u senzoru nebo hudba s normální hlasitostí a 30-100% křik nebo hlasitá hudba.

Použití Sestavení robota reagujícího na zvuk. V NXT-G lze senzor použít k nahrání zvuku, který poté může kostka přehrávat.

4.1.4 Světelný senzor

Světelný senzor (obr. 11) měří intenzitu světla dopadajícího na čidlo. Intenzita je v procentech, přičemž 0% reprezentuje černou a 100% bílou barvu. Pracuje rovněž ve dvou režimech. V prvním pouze pasivně měří intenzitu světla v okolí a ve druhém se rozsvítí červená dioda, takže může podle množství odraženého světla rozpoznat různé barvy. Senzor "vidí" v odstínech šedi, takže nemůže určit o jakou barvu se jedná, pouze je dokáže od sebe odlišit.



Obrázek 11: Světelný senzor

Použití Světelný senzor se nejčastěji využívá při konstrukci robota sledujícího čáru. Může ale také v omezené míře rozlišovat barvy.

4.1.5 Senzor barev



Obrázek 12: Lego Senzor barev

Senzor barev (obr. 12) má tři různé funkce. Umožňuje robotu rozlišovat barvy. Dokáže rozpoznat 6 různých barev. Nemá podporu v RobotC.

Lze jej také použít pro rozlišení světelné intenzity. Má LED, která umí svítit červeně, zeleně a modře. Je-li barva světla nastavena na červenou, funguje jako světelný senzor. Použitím jiné barvy (zelené či modré) se získají jiné výsledky. Senzor lze využít k rozlišení světelné intenzity od rozptýleného světla po odraz světla.

Pro zajištění optimálního rozpoznání barev by měl být senzor držen v pravém úhlu přibližně 1 mm nad povrchem. Nesprávné rozlišení barev se může objevit v případě, že je senzor držen v jiném úhlu vzhledem k povrchu nebo když je používán v jasném osvětlení.

Použití Senzor barev se může využít k roztřídění Lego kostek nebo k sestrojení robota, který bude sledovat barevnou čáru nebo změni směr, když uvidí určitou barvu. Tento senzor lze také využít jako barevnou lampičku.

4.1.6 Tepelný senzor

Senzor je digitální a měří s přesností na 12 bitů. Samotné čidlo je ve špičce dlouhého kovového obalu, který slouží k ochraně před kapalinami (obr 13). Spojovací kabel je napevno připevněný k senzoru.



Obrázek 13: Tepelný senzor

Technické parametry:

- Rozsah: -40 až 125 °C (snese vroucí vodu)
- Přesnost: 0.0625 °C
- Délka čidla: 6.4 cm
- Délka kabelu: 50.5 cm

Použití: Měření teploty.

Poznámka: Senzor je podporován pouze v Educational verzi NXT-G. Pro použití v normální verzi je třeba importovat NXT-G blok. Toto lze ovšem provést pouze u NXT-G v1.0, v NXT-G 2.0 je nefunkční. Lze jej najít na stránkách Team Hassenplug [3]. V RobotC a VPL nemá senzor podporu.

4.2 HiTechnic

V této sekci jsou senzory firmy HiTechnic [1].

4.2.1 Akcelerometr

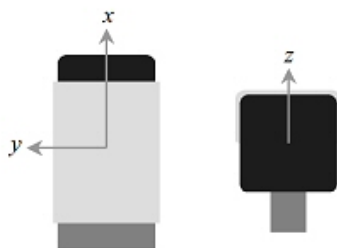


Obrázek 14: Akcelerometr

Akcelerometr obsahuje tři akcelometrické čidla, která měří zrychlení v osách x , y a z . Zrychlení se měří v rozsahu od $-2g$ do $+2g$ s přesností přibližně $0.005g$.

Akcelerometr lze také použít k měření náklonu ve třech osách. Senzor používá digitální I2C komunikační protokol. Měření pro každou osu se opakuje přibližně 100krát za sekundu.

Označení os x , y a z je podle obrázku 15 [1].



Obrázek 15: Osy měření akcelerometru

Použití Měření zrychlení při volném pádu, vytvoření ovládání robota závislého na pohybu senzoru, měření náklonu.

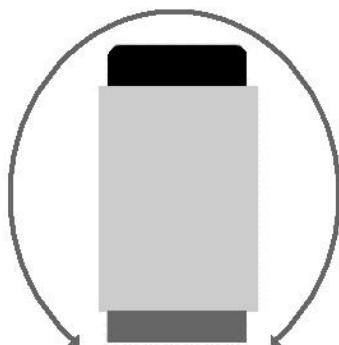
4.2.2 Gyroskopický senzor



Obrázek 16: Gyroskopický senzor

Gyroskopický senzor měří otáčky okolo osy. Vrací hodnotu reprezentující počet stupňů za sekundu otáčení. Čidlo dokáže zaznamenat rotaci až $\pm 360^\circ$ za sekundu otáčení.

Otáčky se čtou přibližně 300krát za sekundu. Čidlo měří ve vertikální rovině orientované černým koncem nahoru jako na obrázku 17. [1]



Obrázek 17: Detekční rovina gyroskopického senzoru

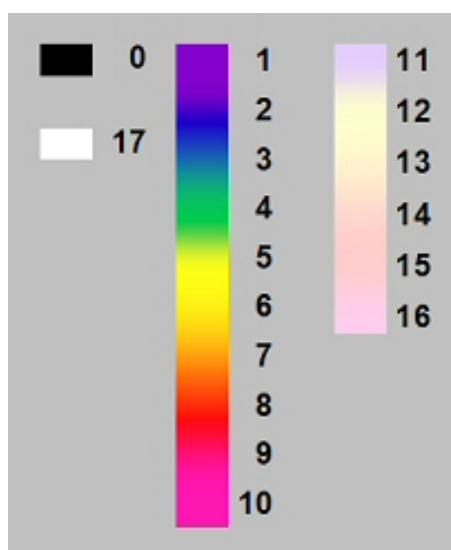
Použití Měření rychlosti otáčení. Neúspěšně jsem se pokoušel vytvořit balancujícího robota, ale našel jsem Implementaci v RobotC [7].



Obrázek 18: HiTechnic senzor barev V2

4.2.3 Senzor barev V2

Nový HiTechnic Senzor barev verze 2 (V2) analyzuje barevné složky odraženého světla a následně vypočítá číselnou hodnotu barvy, kterou vrátí programu. K osvětlení objektů používá jednu bílou LED diodu. Měření opakuje přibližně 100krát za sekundu. Přesnost měření závisí na světelných podmínkách a uhlu, pod kterým senzor snímá. Na obrázku 19 je ukázán vztah mezi číselnou hodnotou barvy a barvou objektu. Senzor používá digitální I2C komunikační protokol.



Obrázek 19: Hodnoty barev

Konfigurace Senzor barev bývá často používán v prostředí osvětleném elektrickými světly. Tyto světla blikají frekvencí jejich elektrického zdroje a toto blikání může senzor ovlivnit. Rušení se lze zbavit konfigurací senzoru.

Senzor barev musí být nakonfigurován na frekvenci rozvodné sítě. Ve světě se běžně používají dvě frekvence: 50 Hz a 60 Hz. Senzor je výrobně nastaven pro státy s 60 Hz rozvodnou sítí, takže v případě použití v takovémto státě jej není nutno konfigurovat. Pokud jej však používáme jinde, je konfigurace nutností. K tomu stačí spustit program, který ke stažení na stránkách výrobce.[1] V České Republice je frekvence rozvodné sítě 50 Hz.

Adresa	Typ	Obsah
00 - 07H	znaky	Číslo verze senzoru
08 - 0FH	znaky	Výrobce
10 - 17H	znaky	Typ senzoru
18 - 3DH	znaky	Nepoužívá se
3E, 3FH	znaky	Rezervováno
40H	byte	Nepoužívá se
41H	byte	Příkaz
42H	byte	Číslo barvy
43H	byte	Hodnota červené složky
44H	byte	Hodnota zelené složky
45H	byte	Hodnota modré složky

Tabulka 1: Rozvržení registru HiTechnic barevného detektoru

Popis registru(tabulka 1):

- *Číslo verze senzoru* je ve formátu "Vn.m", kde n je hlavní číslo verze a m je číslo revize. Číslo verze určuje úroveň hardwaru a číslo revize úroveň firmwaru.
- Pole *Výrobce* obsahuje "HiTechnic".
- *Typ senzoru* obsahuje "ColorPD".
- Pole *Příkaz* je rezervováno pro budoucí použití.
- V poli *Číslo barvy* je číselná hodnota odhadované barvy.
- V položkách *Hodnota červené, zelené a modré složky* je aktuální naměřená hodnota pro každou z barevných komponent.

4.2.4 Kompas

Kompas obsahuje digitální magnetické čidlo, které měří magnetické pole Země a určí úhel pod kterým je senzor otočen vzhledem k severu. Tento úhel se udává číslem od 0 do 359. Hodnoty se zaokrouhlují na 1° a obnovují se 100krát za sekundu. Kompas mimo tohoto absolutního směru lze nastavit na relativní směr. Při tomto nastavení určíme referenční směr, který bude sloužit jako 0. Senzor pak vrací hodnoty od -179° do 180°.



Obrázek 20: Kompas

Adresa	Typ	Obsah
00 - 07H	znaky	Číslo verze senzoru
08 - 0FH	znaky	Výrobce
10 - 17H	znaky	Typ senzoru
18 - 3DH	znaky	Nepoužívá se
3E, 3FH	znaky	Rezervováno
40H	byte	Nepoužívá se
41H	byte	Výběr módu
42H	byte	Směr - dvoustupňový směr
43H	byte	Směr - jednostupňový doplněk
44, 45H	word	Směr (nízký byte, vysoký byte)
46 - 7FH	bytes	Nepoužívá se

Tabulka 2: Rozvržení registru kompasu

Popis registru (tabulka 2)

- Číslo verze senzoru je ve formátu "Vn.m", kde n je hlavní číslo verze a m je číslo revize. Číslo verze určuje úroveň hardwaru a číslo revize úroveň firmwaru.
- Pole Výrobce obsahuje "HiTechnic".
- Typ senzoru obsahuje "Compass".
- Výběr módu může být nastaveno buď na 0x00 nebo 0x43 podle toho, jestli senzor právě měří nebo se kalibruje.
- Pole Směr uchovávají hodnotu právě vypočítaného směru. Během kalibrace se tato pole nedají použít.
- Hodnota směru se získá z adresy 0x42 zjištěním dvoustupňového směru a adresy 0x43 zjištěním jednostupňového doplňku. Tyto hodnoty se poté spočítají rovnicí:

$$\text{Směr} = (\text{dvoustupňový směr} \cdot 2) + \text{jednostupňový doplněk}$$

Kompas funguje správně pouze v tom případě, že je ve vodorovné poloze. Na to je zapotřebí pamatovat při konstrukci robota. Doporučuje se proto poskládat robota tak, aby byl kompas alespoň 15 cm od motorů a 10 cm od NXT kostky. Je nutné se ujistit, že je senzor pevně uchycen. Měření by nemuselo být přesné, kdyby senzor za jízdy poskakoval. [1]

Použití Korekce natočení robota.

4.2.5 EOPD



Obrázek 21: EOPD

EOPD je elektro-optický detektor přiblížení, tedy elektronické zařízení, které používá viditelné světlo ke své funkci. EOPD je podobné jako standardní světelný senzor kromě toho, že používá světelné pulzy k odstranění rušení. Rušení může být způsobeno rozptýleným světlem nebo světlem za překážkou. Senzor zjistí rozdíl hodnot před pulzem a během pulzu. Tento rozdíl se pak odečte od naměřené hodnoty. Tímto je zaručeno, že měření odraženého světla není závislé na okolním světle.

Senzor funguje ve dvou módech: s x1 citlivostí a x4 citlivostí. V módu s x4 citlivostí dokáže EOPD rozeznat míč z Lego Mindstorms NXT sady na vzdálenost alespoň 15 centimetrů. V módu s x1 citlivostí dokáže rozeznat malé předměty.

Z EOPD můžeme číst až 300krát za sekundu. Senzor ale snímá s frekvencí 350 až 400 vzorků za sekundu, aby bylo zajištěno, že NXT bude mít neustále nová data. Každý EOPD senzor funguje s odlišnou vzorkovací frekvencí, což znamená, že můžeme používat více těchto senzorů najednou s minimální šancí, že se budou mezi sebou rušit. [1]

Použití Detekce překážek.



Obrázek 22: IR vyhledávač V2

4.2.6 Infračervený vyhledávač

Infračervený vyhledávač je detektor infračerveného signálu obsahující 5 čidel, které zajišťují příjem signálu z 240 stupňové výseče. Senzor může přijímat signál ze zdrojů jako je Slunce, infračervené dálkové ovládání nebo HiTechnic infračervený míč. Senzor používá pokročilé digitální zpracování k filtraci přijatého a následném výběru pouze toho signálu, který je požadován.

IR vyhledávač funguje ve dvou módech:

- **Modulovaný (AC) mód** - Senzor detekuje modulovaný IR signál, jaký například vysílá HiTechnic Infračervený míč nebo některé dálkové ovládače. V modulovaném módu senzor ignoruje většinu ostatních IR signálů za účelem snížení rušení ze zdrojů, jako je např. slunce nebo světla. Čidlo je nastavené na příjem obdélníkového signálu s frekvencí 1200Hz.
- **Nemodulovaný (DC) mód** - Senzor detekuje nemodulovaný IR signál, který např. vyzařuje slunce nebo starší verze Infračerveného míče.

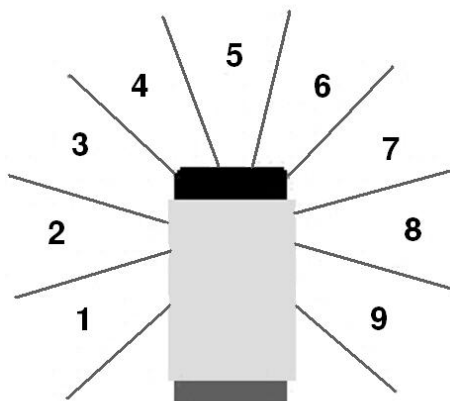
Při použití dálkového ovládání jako zdroje signálu je v AC módu většina ovládačů detekována v závislosti na vlastnostech jejich signálu. IR vyhledávač je nastavený na příjem obdélníkového signálu s frekvencí 1200Hz.

Hodnoty směru, které senzor vrací, jsou od 1 (vlevo vzadu) po 9 (vpravo vzadu). Přičemž 5 znamená, že zdroj je přímo před senzorem, viz. obrázek 23 [1].

Popis registru (tabulka 3)

- *Mód* určuje typ AC signálu, který senzor přijímá (600 Hz, 1200 Hz).
- *DC směr* udává odkud je přijímán nejsilnější DC signál.
- Síly signálu pro každý DC a AC senzor jsou senzorem změřené hodnoty.

Použití Určení směru a měření síly signálu.



Obrázek 23: Hodnoty směru IR vyhledávače

Adresa	Typ	Obsah
41H	byte	Mód
42H	byte	DC směr(0 - 9)
43H	byte	Síla signálu DC senzoru 1
44H	byte	Síla signálu DC senzoru 2
45H	byte	Síla signálu DC senzoru 3
46H	byte	Síla signálu DC senzoru 4
47H	byte	Síla signálu DC senzoru 5
48H	byte	Průměrná síla DC signálu
4AH	byte	Síla signálu AC senzoru 1
4BH	byte	Síla signálu AC senzoru 2
4CH	byte	Síla signálu AC senzoru 3
4DH	byte	Síla signálu AC senzoru 4
4EH	byte	Síla signálu AC senzoru 5

Tabulka 3: Rozvržení registru IR vyhledávače

4.3 Příslušenství

K NXT lze dokoupit také řadu příslušenství, které rozšiřují možnosti využití stavebnice. K těmto doplňkům například patří sensorový multiplexer, který zvětší počet vstupních portů na senzory, dálkové ovládače, adaptéry na starší verze Lego Mindstorms.

4.3.1 Hitechnic Infračervený míč

HiTechnic Infračervený míč je ideální pro robotický fotbal nebo jako IR navádění pro roboty. Je vyvážený a obsahuje 20 infračervených LED diod pro maximální pokrytí



Obrázek 24: Infračervený míč

signálem. Může fungovat v modulovaném a nemodulovaném režimu pro soutěž Robocup Jr a ve dvou HiTechnic režimech, které vysílají na frekvencích 1200Hz a 600Hz. Tyto HiTechnic režimy vylepšují dosah signálu o více než 5 metrů a prodlužují životnost baterií. Pulzně modulované signály usnadňují detekci IR míče za náročných světelných podmínek. Míč má průměr 75mm a napájí jej 4 AAA baterie[1].

4.4 Porovnání senzorů

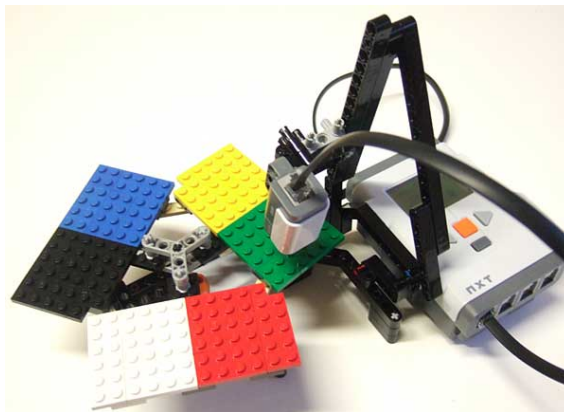
4.4.1 Akcelerometr, gyroskop a kompas

Akcelerometr, kompas a gyroskop jsou senzory, které měří veličiny závislé na pohybu. Tyto hodnoty a s tím i použití senzorů se liší. Rozdíl mezi těmito třemi senzory je následující. Rychlost je změna pozice (jak rychle se robot pohybuje z bodu A do bodu B) a zrychlení je změna rychlosti (za jak dlouho robot z pomalého pohybu přejde do rychlého).

- Akcelerometr měří **lineární zrychlení**, tzn. změnu rychlosti. Dokáže také využít naměřenou gravitaci k detekci náklonu. Gravitace je zrychlení směrem ke středu Země. Z akcelerometru umístěného v klidu na vodorovné podložce odečteme hodnotu 1g směřující přímo kolmo k podložce. V případě, kdy hodnota 1g směřuje do strany je senzor nakloněn. Akcelerometr nedokáže zjistit otáčky kolem svého středu.
- Gyroskop měří **úhlovou rychlost** ne zrychlení. Hodnota "0", naměřená gyroskopem, znamená, že se tento senzor kolem své osy netočí. Pokud je naměřena hodnota "x", znamená to, že se otáčí o "x" stupňů za sekundu.
- Kompas měří své **natočení vzhledem k severu**. Lze jej využít i k měření itáček kolem středu senzoru.

4.4.2 Barevný detektor Lego a HiTechnic

Senzory fungují na odlišném principu[10]. Lego senzor používá RGB LED diodu a postupně svítí na objekt červené, zelené a modré světlo. Odražené světlo měří světelné čidlo citlivé na všechny vlnové délky. Naopak HiTechnic senzor má jednu bílou LED diodu a specializovaný čip citlivý na barvu. Tento čip má tři citlivé oblasti pokryté červeným, zeleným a modrým filtrem. Výhoda této metody je, že dokáže zjistit barvu vstupujícího světla i bez zapnutí LED diody.



Obrázek 25: Model na test oblasti detekce

Test oblasti detekce Tento test se prováděl na modelu, který je na obrázku 25. Šest barevných destiček je přiděleno na NXT motor, který s nimi otáčí a senzor snímá barvu. Měření se opakuje 10krát v každé vzdálenosti. Pro zjednodušení se test prováděl s šesti barvami, které je Lego senzor schopen v NXT-G rozlišit. Za zmínku stojí, že HiTechnic dokáže v NXT-G rozlišit 18 barev a zjistit barevné složky. Přístup k barevným složkám u Lego senzoru je možné použitím jiného programovacího jazyka.

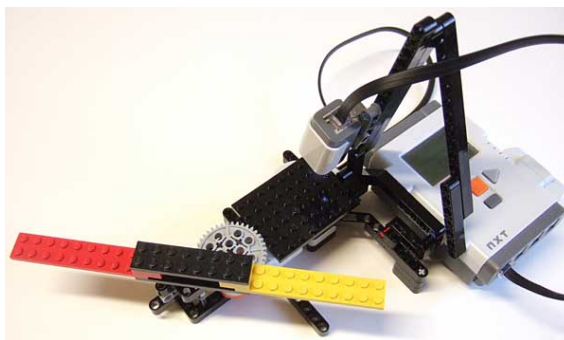
Temná místnost, senzor barev kolmo k destičkám Zkušební vzdálenost mezi čidlem senzoru a destičkami je 1, 2, 3, 5 a 8 dírek na vertikálním dílku, který senzor drží. Lego senzor spolehlivě poskytuje hodnoty ve vzdálenosti 1 a 2 dírky. Na 3 dírkách přečte zelenou jako černou. Na 5 a 8 jsou už všechny barvy černé.

Při stejném postupu vrací HiTechnic senzor nepřesné hodnoty např. černou rozezná jako bílou. Toto může být způsobeno odrazem ze světelného zdroje senzoru a přehlcním senzoru.

Temná místnost, senzor barev mírně nakloněný V druhém testu byl senzor nakloněný asi o 10°. Lego senzor se chová stejně jako v předchozím měření. HiTechnic ale v tomto případě měří přesně až do vzdálenosti 5 dírek a na 8 dírkách rozezná správně všechny barvy až na zelenou, kterou vidí jako černou.

Osvětlená místnost, senzor barev mírně nakloněný Další měření testuje závislost na osvětlení. Při použití stejného modelu a 35W halogenové lampy umístěné asi 55cm od destiček. Tentokrát Lego čte správně pouze ve vzdálenosti 1 dírky, na ostatní vzdálenosti vrací černou barvu. HiTechnic senzor není osvětlením vůbec ovlivněn a výsledky jsou stejné, jako v tmavé místnosti.

Denní světlo, senzor barev mírně nakloněný Při tomto měření je model umístěn na sluneční světlo za mírně zataženého dne. Za těchto podmínek je Lego senzor nepoužitelný už na nejmenší vzdálenosti. HiTechnic zůstává světlem neovlivněný. Při zkoušce na přímém slunečním světle už ale není schopen podat spolehlivé výsledky.



Obrázek 26: Model na test rychlosti čtení

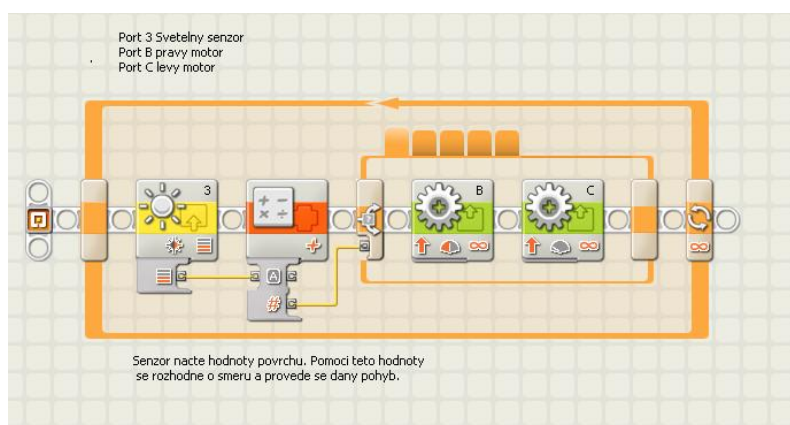
Zkouška rychlosti Tento test je na rychlost čtení. K tomu byl použit model na obrázku 26. Červeno-žlutá vrtule se může před senzorem otáčet různou rychlostí a při znalosti jejího průměru, šířky rychlosti otáček není složité zjistit, jak dlouho je list před senzorem. NXT-G program počítá kolikrát byla rozeznána červená a žlutá, Toto číslo se porovná s počtem otáček vrtule. Rychlost se zvyšuje tak dlouho dokud se měření nestane nespolehlivým. V této zkoušce je lepší Lego senzor s rychlostí 2.5 milisekundy na rozeznání listu. Naopak HiTechnic potřebuje 17 milisekund na spolehlivou detekci barvy. Tento rozdíl je pravděpodobně způsoben jiným spojením s NXT kostkou. Lego senzor se přímo napojuje na NXT ARM mikropočítač pomocí A/D převodníku, zatímco HiTechnic používá pomalejší I2C rozhraní.

Závěr Každý senzor má své silné stránky a hodí se nejlépe pro jiné případy. Lego senzor barev se uplatní za situace, kdy je důležitá rychlost, jako je sledování barevné čary. HiTechnic senzor zase dokáže rozeznat barvy za náročných světelných podmínek a umí lépe rozlišovat mezi barevnými odstíny.

5 Příklady použití

Pro otestování senzoru jsem naimplementoval několik příkladů, které demonstrují vlastnosti senzorů. Modely robotů jsou založeny na modelu Five minute bot [11] s drobnými úpravami. Návod na složení je spolu se zdrojovými kódy a videoukázkami na přiloženém CD.

5.1 Světelný senzor



Obrázek 27: Sledování čáry NXT-G

Popis

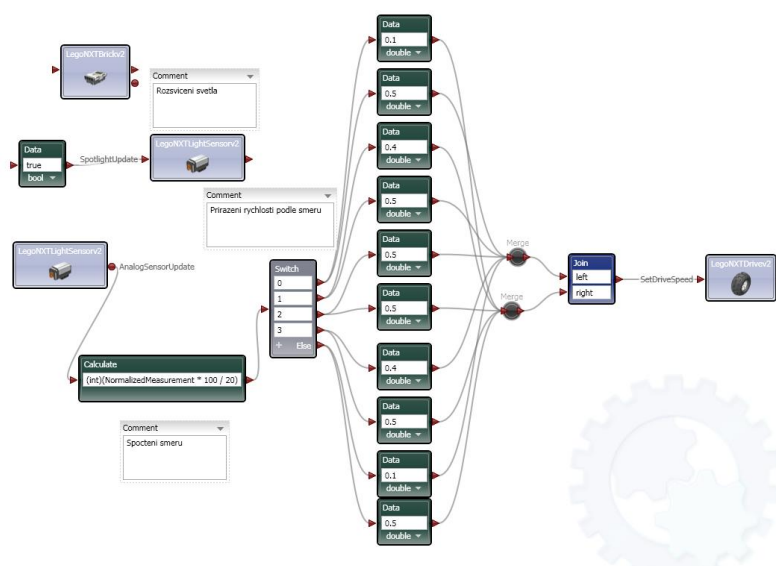
Robot pomocí světelného senzoru sleduje čáru. Naimplementováno v RobotC , NXT-G, a VPL (viz. obr. 27 a 28).

Princip

Před samotným použitím je potřeba kalibrovat světelný senzor na aktuální světelné podmínky. Robot sleduje čáru opakovaným otáčením doleva a doprava. Při použití černé čáry na bílém povrchu by světelný senzor naměřit na hraně hodnotu okolo 50.

Na hodnoty senzoru reaguje robot následujícím způsobem:

- Hodnota mezi 0-20: ostrá otáčka do prava.
- Hodnota mezi 21-40: jemná otáčka do prava.
- Hodnota mezi 41-60: jede rovně.
- Hodnota mezi 61-80: jemná otáčka do leva.



Obrázek 28: Sledování čáry VPN

- Hodnota mezi 81-100: ostrá otáčka doleva.

Tento přístup způsobuje, že robot jezdí "cik-cak". Kvalita algoritmu se posuzuje podle plynulosti, jakou sleduje čáru.

```
#pragma config(Sensor, S3, lightSensor, sensorLightActive)
#pragma config(Motor, motorB, left, tmotorNormal, openLoop, encoder)
#pragma config(Motor, motorC, right, tmotorNormal, openLoop, encoder)
// *!!Code automatically generated by 'ROBOTC' configuration wizard!!*//

#include "drivers/LEGOLS-driver.h"

task main
{
    while(true)
    {
        int reading=LSvalNorm(lightSensor)/20; //cteni normalizovane hodnoty po kalibraci
        switch(reading) // nastaveni rychlosti motoru v zavislosti na prectene hodnotě
        {
            case 0: motor[left]=50; motor[right]=10; break;
            case 1: motor[left]=50; motor[right]=40; break;
            case 2: motor[left]=50; motor[right]=50; break;
            case 3: motor[left]=40; motor[right]=50; break;
            default: motor[left]=10; motor[right]=50; break;
        }
    }
}
```

Výpis 2: Verze bez kalibrace v RobotC

5.2 IR vyhledávač, IR míč

Popis

Robot demonstruje použití IR vyhledávače při detekci IR míče. Naimplementováno v RobotC

Princip

IR vyhledávač je nastaven na detekci signálu s frekvencí 1200 Hz, který vysílá IR míč. Robot se v závislosti na směru odkud senzor přijímá signál otočí tak, ať směřuje čelem k míči. Poté se k němu přiblíží. Jestliže se míč pohybuje, robot jej sleduje.

```
#pragma config(Sensor, S3, IRSeeker, sensorHiTechnicIRSeeker1200)
#pragma config(Sensor, S4, Touch, sensorTouch)
#pragma config(Motor, motorB, motorRight, tmotorNormal, openLoop, encoder)
#pragma config(Motor, motorC, motorLeft, tmotorNormal, openLoop, encoder)
// **!!Code automatically generated by 'ROBOTC' configuration wizard **!!
```

```
/* Robot pomoci IR vyhledavace urci zdroj signalu a jede jeho smerem.*/
```

```
#include "drivers\common.h"
#include "drivers\HTDIR-driver.h"
```

```
int dir; //smer signalu
```

```
// main task
```

```
task main ()
```

```
{
```

```
wait1Msec(2000);
```

```
PlaySound(soundBeepBeep);
```

```
while(bSoundActive);
```

```
//nastaveni modu vyhledavace
```

```
HTDIRsetDSPMode(S3,DSP_1200);
```

```
//prommena urcujici rychlost motoru
```

```
int power = 70;
```

```
int rychlostB, rychlostC; // rychlosti praveho a leveho motoru
```

```
while(true)
```

```
{
```

```
//smeru signalu
```

```
dir = HTDIRreadACDir(IRSeeker);
```

```
// zobrazeni smeru (hodnoty -5 az 4, 0 znamena ze cil je pred robotem)
```

```
nxtDisplayTextLine(0,"smer:_%d", dir-5);
```

```
if (dir == 5)
```

```
{
```

```
motor[motorRight]=power;
```

```

        motor[motorLeft]=power;
    }
    // proporcionalni urceni rychlosti motoru
    else{
        rychlostB=power-((dir-5)*30);
        rychlostC=power+((dir-5)*30);
        motor[motorRight]=rychlostB;
        motor[motorLeft]=rychlostC;
    }
}
}

```

Výpis 3: Zdrojový kód příkladu použití IR vyhledávače

5.3 EOPD, Zvukový a Ultrazvukový senzor

Popis

Robot pomocí EOPD senzoru sleduje stěnu na své pravé straně a ultrazvukový senzor detekuje vnitřní rohy [12]. Při detekci hlasitého zvuku se robot zastaví. Nainplementováno v NXT-G

Princip

Robot sleduje stěnu a případně podle hodnoty EOPD provádí korekce vzdálenosti od stěny. Pokud je robot příliš blízko jemně zatočí do leva, pokud daleko tak do prava. V případě, že ultrazvukový senzor zaznamená blízkou stěnu, tak se ji robot vyhne odbočením do leva.

5.4 Akcelerometr a Dotykový senzor

Popis

Pomocí naklánění akcelerometru ovládáme pohyb robota, motory se spouští stisknutím dotykového senzoru. Nainplementováno v RobotC

Princip

Jak bylo řečeno u popisu akcelerometru, tento senzor dokáže zjistit svou polohu. Tohoto využijeme při ovládání robota. Při naklonění ovládače s akcelerometrem se díky gravitaci změní hodnoty zrychlení jednotlivých os. Tyto hodnoty, ale pořád určují směr dolů. Díky tomu jsme schopni zjistit, na kterou stranu je ovládač nakloněn a tomu přizpůsobíme pohyb robota. Robot se rozjede dopředu, pokud nakloníme akcelerometr dopředu. Dozadu bude couvat a do stran zatáčet.

Zdrojový kód je v příloze v kapitole B.1.

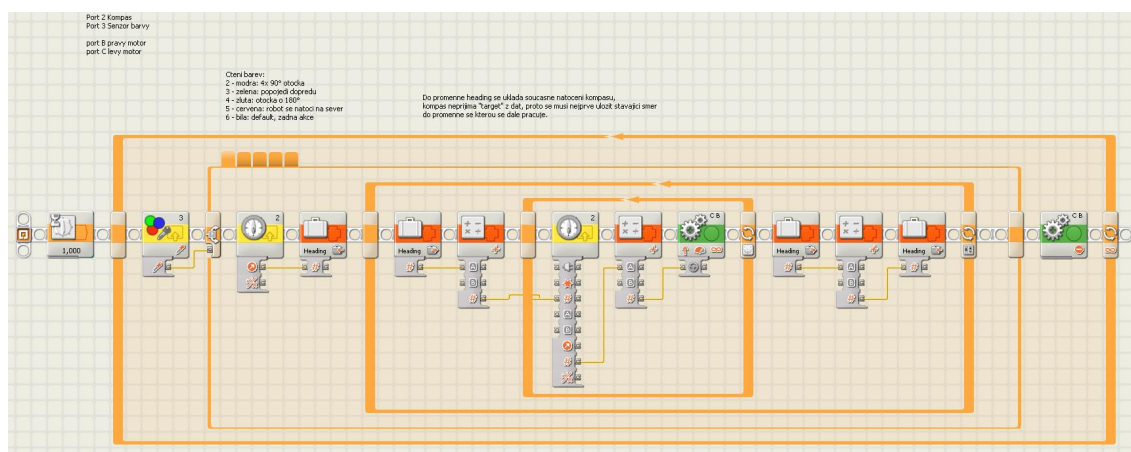
5.5 Senzor barev a Kompas

Popis

Robota ovládáme pomocí barev. Při detekci určité barvy provede robot s pomocí kompasu odpovídající pohyb. Naimplementováno v NXT-G.

Princip

- **červená:** Robot se nasměruje na sever.
- **žlutá:** Robot se otočí o 180° .
- **modrá:** Robot se otočí o 90° a popojede dopředu. Toto opakuje 4x, takže se vrátí na stejnou pozici, ze které vyjel.
- **zelená:** Robot popojede dopředu.



Obrázek 29: Příklad na použití kompasu a detektoru barev

6 Závěr

Prostřednictvím této bakalářské práce jsem se seznámil se stavebnicí Lego Mindstorms NXT a robotikou obecně a své poznatky a postřehy jsem zde shrnul. Na tvorbě robotů v NXT jsou inspirující jejich téměř neomezené možnosti, které limituje pouze fantazie a dovednost tvůrce.

Z programovacích jazyků popsaných v této práci se jeví uživatelsky nejvýhodnější RobotC. U NXT-G jsem zaznamenal několik limitujících vlastností, kvůli kterým bych jej nedoporučil. Z bloků nelze na první pohled vyčíst hodnoty, ty jdou zjistit až při označení bloku. Toto dělá program nepřehledným a komplikuje práci. U VPL mi nevyhovovalo zpoždění způsobené připojením přes bluetooth.

V práci jsem popsal stavebnici Lego Mindstorms NXT, její parametry a využití. Dále jsem otestoval senzory Lego a HiTechnic. K těmto senzorům jsem s výjimkou gyroskopu napsal vzorové programy. Poznatky a postřehy, které jsem zde uvedeny mohou posloužit jako vodítko pro správný výběr senzorů a vývojového prostředí dle konkrétních požadavků uživatele na výsledné vlastnosti robota.

Téma mi bylo velice blízké a mnohdy jsem plnění úkolů, obsažených v této práci vnímal spíše jako hru. Do budoucna by bylo zajímavé vytvořit si vlastní implementaci balancujícího robota.

7 Reference

- [1] Hitechnic[online]. 2009 [cit. 2010-05-02]. *Hitechnic Products*. Dostupné z WWW: <<http://www.hitechnic.com/products>>.
- [2] HiTechnic[online]. 2009 [cit. 2010-05-02]. *Hitechnic - Download*. Dostupné z WWW: <<http://http://www.hitechnic.com/downloads>>.
- [3] Team Hassenplug [online]. 2010 [cit. 2010-05-02]. *Team Hassenplug*. Dostupné z WWW: <<http://www.teamhassenplug.org/NXT/NXTGAdditions.html>>.
- [4] Team Hassenplug [online]. 2010 [cit. 2010-05-02]. *NXT Programming Software*. Dostupné z WWW: <<http://www.teamhassenplug.org/NXT/NXTSoftware.html>>.
- [5] ROBOTC.net [online]. 2010-04-26 [cit. 2010-05-02]. *ROBOTC.net*. Dostupné z WWW: <<http://www.robotc.net/>>.
- [6] Lego Education: Store [online]. 2010 [cit. 2010-05-02]. *Lego Education: Store*. Dostupné z WWW: <<http://www.legoeducation.us/store/>>.
- [7] The NXT STEP [online]. 2010 [cit. 2010-05-02]. *Robot Inspiration Series #3: Self-Balancing Wheeled Robots - Part 3*. Dostupné z WWW: <http://thenxtstep.blogspot.com/2008/07/robot-inspiration-series-3-self_23.html>.
- [8] MRDS Services for HiTechnic [online]. 2010-02-08 [cit. 2010-05-02]. *MRDS Services for HiTechnic*. Dostupné z WWW: <<http://mrdshitechnic.codeplex.com/>>.
- [9] Visual Programming Language [online]. 2010 [cit. 2010-05-02]. *Visual Programming Language*. Dostupné z WWW: <[http://msdn.microsoft.com/en-us/library/bb964572\(v=MSDN.10\).aspx](http://msdn.microsoft.com/en-us/library/bb964572(v=MSDN.10).aspx)>.
- [10] Philo's Home Page [online]. 2010 [cit. 2010-05-02]. *Color Sensor Showdown*. Dostupné z WWW: <<http://www.philohome.com/colcomp/cc.htm>>.
- [11] NXT Programs [online]. 2010 [cit. 2010-05-02]. *NXT Five Minute Bot*. Dostupné z WWW: <http://nxtprograms.com/five_minute_bot/index.html>.
- [12] HiTechnic [online]. 2010 [cit. 2010-05-02]. *Wall bot - an EOPD Based Wall Follower*. Dostupné z WWW: <<http://www.hitechnic.com/blog/eopd-sensor/wallbot/>>.

A Obsah CD

- text bakalářské práce
- zdrojové kódy
- návody na sestavení
- videa příkladů použití z kapitoly 5.

B Zdrojové kódy

B.1 Akcelerometr a dotykový senzor

```

#pragma config(Sensor, S3, accel, sensorI2CHiTechnicAccel)
#pragma config(Sensor, S4, touch, sensorTouch)
#pragma config(Motor, motorB, right, tmotorNormal, PIDControl, encoder)
#pragma config(Motor, motorC, left, tmotorNormal, PIDControl, encoder)
// *!! Code automatically generated by 'ROBOTC' configuration wizard !!*//

/* robot ovladany pozici akcelerometru*/

#include "drivers/common.h"
#include "drivers/HTAC-driver.h"

task main () {
    int x_axis = 0; /*hodnota zrychleni osy X, kladna hodota – zakloneny senzor,
                    zaporna hodnota – predkloneny senzor*/

    int y_axis = 0; /*hodnota zrychleni osy Y, kladna hodota – senzor nakloneny do prava,
                    zaporna hodnota – senzor nakloneny do leva*/

    int LPow = 0; //rychlost leveho motoru
    int RPow = 0; //rychlost praveho motoru

    wait1Msec(2000);
    PlaySound(soundBeepBeep);
    while(bSoundActive);

    while (true) {
        // motory bezi pouze kdyz je sepnuty dotykový senzor
        if (SensorValue[touch]){

            LPow=0;
            RPow=0;

            // precteni zrychleni jednotlivých os
            HTACreadX(accel, x_axis);
            HTACreadY(accel, y_axis);

            //zakloneni
            if (x_axis>100)
            {
                LPow—=(x_axis—100);
                RPow—=(x_axis—100);
            }

            //nakloneni do prava
            if (y_axis>100)
            {

```



```
        LPow+=(y_axis-100);
        RPow-=(y_axis-100);
    }

    //predkloneni
    if (x_axis<-100)
    {
        LPow-= (x_axis+100);
        RPow-= (x_axis+100);
    }

    //nakloneni do leva
    if (y_axis<-100)
    {
        LPow+=(y_axis+100);
        RPow-=(y_axis+100);
    }

    motor[right] = RPow;
    motor[left] = LPow;
}
else
{
    motor[right] = 0;
    motor[left] = 0;
}
}
```

Výpis 4: Zdrojový kód od příkladu s použitím akcelerometru a dotykového senzoru